
P.V.P.
375
48K
Spectrum
Nº 2
sinclair
AÑO I
REVISTA
MENSUAL
COLECCIONABLE
MONSER
PROGRAMA
DEL MES
PANZER
3 D
II PARTE
DEL CURSO
CODIGO
MAQUINA
PROGRAMA
BASIC
CARGAEX
ANALISIS
SOFTWARE
UGH
PERIFERICOS
ANALIZAMOS
LAPIZ OPTICO
REGALO:
UN KEY-PANEL
TODOS LOS MESES
M. NOVOA
CONCURSO
80.000 PTAS EN PREMIOS
BASES EN EL INTERIOR



AÑO I - Nº 2 - MAYO 1985

DIRECTOR
José Nieto Rubio

COORDINADOR
Félix Santamaria

SUPERVISOR SOFTWARE
Gustavo Cano Muñoz

DISEÑO
Angélica Arce

REDACCION
César Hernández
Javier González
H. Carretero
Agustín Barcos
Belén Sánchez Vicente

COLABORADORES
Mario Alvarez
Juncal Feijoo
María Amaya

PORTADA
Mauro Novoa

EDITA
MONSER, S.A.

DIRECTOR EDITORIAL
J. L. Cano Regidor

REDACCION, ADMINISTRACION Y PUBLICIDAD
Argos, 9
28037 MADRID
Tlf. 742 72 12 / 96

PUBLICIDAD Y SUSCRIPCIONES
Yolanda Bardillo

FOTOCOMPOSICION
FERMAR

FOTOMECANICA
IMAGEN
Nicolás Morales, 34-3º

IMPRIME
GRAFICAS G. ABAD
Virgen del Lluc, 73

DISTRIBUCION
DISPRENSA
Eduardo Torroja, 9

Depósito Legal: M.-10328-1985

Se solicitará control O.J.D.

SUMARIO

CONCURSO

80.000 ptas. en premios

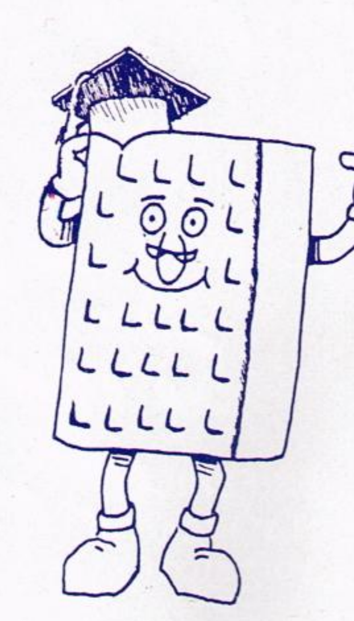
48K abre sus puertas a tus programas, envíanoslos antes del 31 de mayo, vale cualquier programa que sea original es decir que no se haya publicado nunca, el tema consiste también en algo original, deberá de ser un programa lúdico-educacional, es decir, que enseñe y divierta, eso tan manido como teórico, de aprender jugando.

Los premios serán:

1º una colección de 20 unidades de TUS JUEGOS SINGLE en lujoso estuche individual y 10.000 pts. en metálico.

2º una colección de 20 unidades de TUS JUEGOS SINGLE en lujoso estuche individual.

3º una colección completa de TUS JUEGOS 4: 20 juegos en 5 estuches de lujo.



Los envíos, recordar, antes del día 31-V-85, en una cassette, acompañada del listado del juego, e instrucciones completas de carga y uso, a:

48K
c/ ARGOS, 9
28037 MADRID

Esperamos Vuestros Programas.

LAPIZ OPTICO

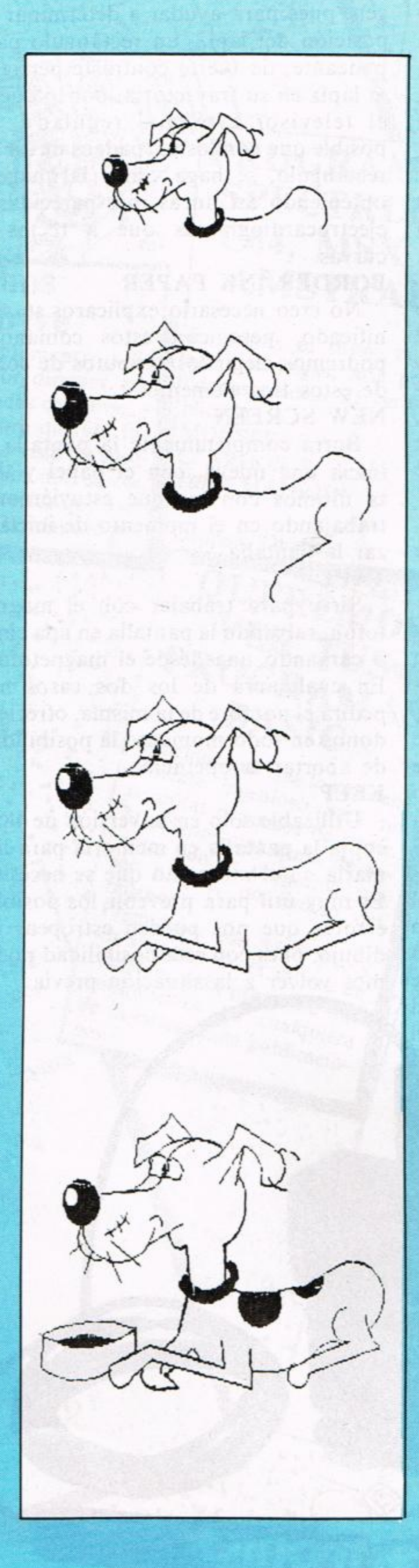
Aunque os prometí, o casi, dibujar "marcianitos" con el lápiz, como veréis no me ha sido posible, principalmente porque no tengo video y los "lagartones" de "V" nunca se están quietos. Así que recurrí a mi ex-mejor amigo (mi chucho) y le dibujé. Lo de "ex" es porque desde que vio el resultado y me mordió, no me ha vuelto a dirigir la palabra.

Bromas aparte, intentaré entrar en materia. El lápiz óptico es un invento capaz de detectar los haces de electrones que "pasan" por la pantalla de televisión y emitir una señar al ordenador. En líneas generales, la imagen del televisor se forma por la reflexión en la pantalla de fósforo de un punto de luz, que recorre la misma a lo largo y ancho, a una velocidad increiblemente rápida. Siendo capaces de medir el instante preciso en que el haz de luz pasa por delante del lápiz, se puede determinar con total exactitud la posición del mismo.

El aparato analizado este mes (LIGHT PEN de DK'TRONICS) a través del programa que le acompaña, utiliza la información recibida desde el lápiz para calcular las coordenadas, X (horizontal) Y (vertical) que determinan su posición exacta en el mapa de pantalla.

Una vez conectada la "interface" que acompaña al lápiz con las precauciones habituales, que como sin duda ya sabeis, consiste en acoplar la "interface" con el SPECTRUM desconectado, se enchufa el ordenador y se carga el programa correspondiente. A propósito, según el fabricante, no es necesario desconeztar la "ZX PRINTER" ni la "INTERFACE 1". Antes de todas estas operaciones se habrá conectado el lápiz a la "interface".

Cuando esté todo listo, el lápiz debe sujetarse firmemente contra la pantalla del televisor, para que la señal de luz recibida sea suficientemente nítida y además evitar reflejos que puedan interferir en la operatividad del lápiz. No es necesario "romper" la pantalla, simplemente apoyarlo sin inclinación. Si alguno de vosotros llega a poseer un artilugio de estos, le deseo más suerte



de la que he tenido yo (al intentar convencer a mis padres que el lápiz no dañaba al televisor "grande") y no tengais que utilizarlo en plan "chorizo" con nocturnidad y a escondidas, pues acaba resultando nocivo para la salud, sobre todo combinado con el... redactor jefe, metiéndote prisas. ¡Qué bronca! por el "tirano" que se me escapó el mes pasado, fue terrible, no hubo forma de convencerle que fue una errata de imprenta, yo que tengo como modelo a mi queridísimo, amadísimo y nunca bien ponderado jefe de redacción.

Perdón, creo que estábamos hablando de lo que se hace una vez cargado el programa y con todo listo para trabajar con el lápiz; en una de sus modalidades para elegir de un menú; o para dibujar ayudado del programa que acompaña al aparato, en la otra.

La elección de un menú se logra a partir de la determinación del nº de la línea en que está el lápiz, pudiendo recurrir a una subrutina del ya mencionado programa, que nos devolverá el nº de línea tan pronto como el lápiz detecte luz. Las instrucciones del aparato, explican exhaustivamente como disponer de ésta y de todas las subrutinas que veremos, incluso como probarlas.

Por último, para dibujar se utiliza un programa cuyas subrutinas nos permitirán realizar cualquier dibujo a través de las 16 opciones, seleccionables del "display", presentado en las dos líneas inferiores de la pantalla.

Antes de pasar a explicar estas opciones es conveniente saber que siempre se necesita, a la hora de dibujar cualquier figura, definir dos, y a veces tres, coordenadas. La primera, que llamaré ORIGEN, es colocada por el ordenador y la segunda FIN, es situada con el lápiz al ponerlo contra la pantalla y presionar una tecla. Estas dos coordenadas nos darán las esquinas de un rectángulo; el centro y radio de una circunferencia; principio y fin de una recta, etc.

A continuación veremos las 16 opciones:



ERASE

Permite borrar la última figura dibujada. Debe tenerse en cuenta que si se cambia la posición del ORIGEN o del FIN posiblemente no se pueda borrar.

DRAW

Dibuja una línea uniendo el ORIGEN y el FIN.

MOVE

Mueve el ORIGEN a la posición de FIN, ya que este último es el único punto que podemos mover con el lápiz.

CIRCLE

Se explica por sí mismo, primero se coloca el ORIGEN en lo que será el centro de la circunferencia, para esto será necesario mover el FIN y a continuación con el comando MOVE, colocar el ORIGEN. Después se colocará el FIN en el punto por donde deba pasar la circunferencia.

RECTANGLE

Dibuja un rectángulo. Para ello debe colocarse el ORIGEN y el FIN en las esquinas opuestas (en diagonal).

FILL

Sirve para rellenar una figura con cualquiera de los ocho colores, incluyendo blanco y negro. Sobre este comando, diremos que es evidente que con anterioridad a la operación de colorear, se deberá haber seleccionado uno de los colores. Os contaré un pequeño accidente que me sucedió, la figura debe estar perfectamente delimitada, pues si se deja algún resquicio, por pequeño que sea, se pueden rellenar figuras próximas e incluso toda la pantalla. Una de las veces que esto me sucedió, llegó a "cuajarse" el programa y tuve que desconectar el Spectrum. Por último para rellenar una figura se debe colocar el FIN en la misma y seleccionar el comando FILL.

HAND-DRAW

Permite dibujar a mano alzada, para ello es muy importante que el televisor esté en buen estado y no vibre la imagen, pues para ayudar a determinar la posición del lápiz, un rectángulo parpadeante, de fuerte contraste persigue al lápiz en su trayectoria, por lo que si el televisor está mal regulado, es posible que con los parpadeos de dicho rectángulo, se haga vibrar la imagen, obteniendo así líneas más parecidas a electrocardiogramas que a rectas o curvas.

BORDER INK PAPER

No creo necesario explicaros su significado, pero con estos comandos podremos elegir los atributos de color de estos tres elementos.

NEW SCREEN

Borra completamente la pantalla, e inicia una nueva, con el papel y tinta últimos con los que estuviéramos trabajando en el momento de inicializar la pantalla.

TAPE

Sirve para trabajar con el magnetofón, salvando la pantalla en una cinta o cargando una desde el magnetofón. En cualquiera de los dos casos nos pedirá el nombre de la misma, ofreciéndonos en todo momento la posibilidad de abortar la operación.

KEEP

Utilizable sólo en la versión de 48K, copia la pantalla en memoria para llamarla en el momento que se necesita. Es muy útil para prevenir los posibles errores que nos pueden estropear un dibujo, pues con esta posibilidad podemos volver a la situación previa.

RECALL

Recupera las pantallas almacenadas gracias al comando anterior, presentando un menú con el número de pantallas almacenadas (cinco como máximo)

ARC

Como su nombre casi indica, dibuja arcos o circunferencias, y es una de las opciones que necesitan la fijación de tres puntos como os decía antes:

a) ORIGEN: define un extremo del arco, podríamos llamarle el principio.

b) FIN: es el otro extremo o final.

c) Viene determinado por la posición previa que ocupó el FIN, y define el punto por donde va a pasar el arco.

LETTERS

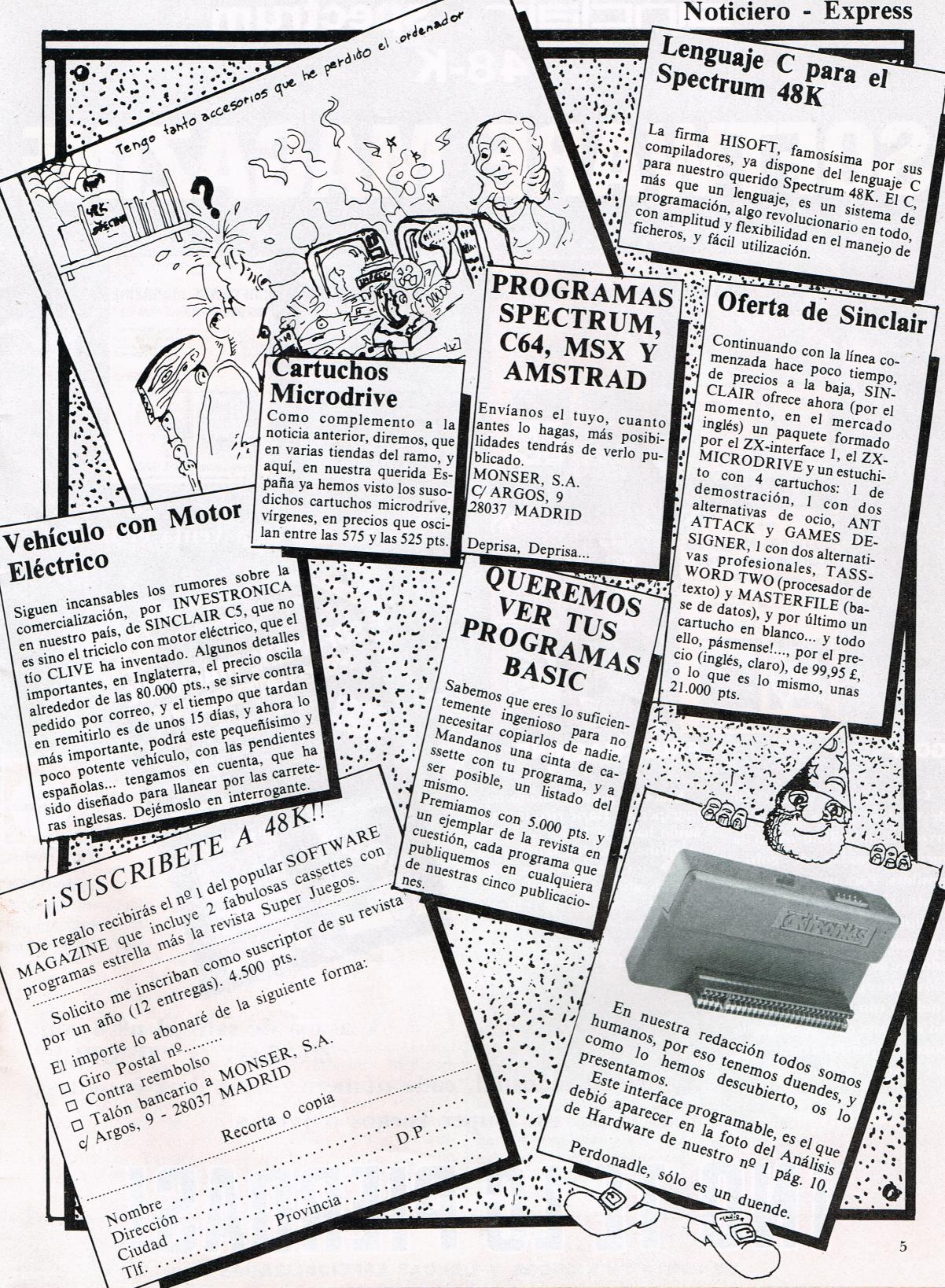
Lo utilizaremos para insertar cualquier tipo de texto; letras; números; incluso caracteres predefinidos por el usuario. El texto se insertará en la posición que marque la coordenada, que hemos llamado FIN.

Por último, las instrucciones del programa, por cierto muy completas, nos indican como utilizar su parte en código máquina independientemente del cargador BASIC, dando una serie de posibilidades muy interesantes, que comprenden, desde salvar el programa completo, una vez calibrado el lápiz, hasta utilizar rutinas que nos devuelvan a un posible programa BASIC el valor de las coordenadas fijadas por el lápiz, para utilizarlos con las instrucciones de la máquina PLOT, DRAW, CIRCLE, o POINT.

A continuación el "manualillo" dará una serie de errores posibles, que se pueden producir utilizando el programa, y la forma de retornar al código-máquina en el hipotético caso de que algún error produjese el retorno al BASIC.

He dejado para lo último el calibrado del lápiz óptico, que supone el ajuste perfecto entre lo que señalamos con el lápiz y lo que aparece en el televisor. Esto se consigue accediendo a una rutina a través del menú principal que nos varía la linealidad horizontal, simplemente presionando un par de teclas. Algunas veces esto no será necesario, pero otras veces, a pesar de esta opción, el ajuste no será todo lo óptimo que se necesite. Si esto sucede, tenemos la última posibilidad, consistente en alterar ciertas constantes del programa en código máquina que calculan el "timing" del haz-electrónico.





Lenguaje C para el Spectrum 48K

La firma HISOFT, famosísima por sus compiladores, ya dispone del lenguaje C para nuestro querido Spectrum 48K. El C, más que un lenguaje, es un sistema de programación, algo revolucionario en todo, con amplitud y flexibilidad en el manejo de ficheros, y fácil utilización.

Cartuchos Microdrive

Como complemento a la noticia anterior, diremos, que en varias tiendas del ramo, y aquí, en nuestra querida España ya hemos visto los susodichos cartuchos microdrive, vírgenes, en precios que oscilan entre las 575 y las 525 pts.

PROGRAMAS SPECTRUM, C64, MSX Y AMSTRAD

Envíanos el tuyo, cuanto antes lo hagas, más posibilidades tendrás de verlo publicado.
MONSER, S.A.
C/ ARGOS, 9
28037 MADRID

Deprisa, Deprisa...

Oferta de Sinclair

Continuando con la línea comenzada hace poco tiempo, de precios a la baja, SINCLAIR ofrece ahora (por el momento, en el mercado inglés) un paquete formado por el ZX-interface 1, el ZX-MICRODRIVE y un estuchito con 4 cartuchos: 1 de demostración, 1 con dos alternativas de ocio, ANT ATTACK y GAMES DESIGNER, 1 con dos alternativas profesionales, TASSWORD TWO (procesador de texto) y MASTERFILE (base de datos), y por último un cartucho en blanco... y todo ello, pásmense!..., por el precio (inglés, claro), de 99,95 £, o lo que es lo mismo, unas 21.000 pts.

Vehículo con Motor Eléctrico

Siguen incansables los rumores sobre la comercialización, por INVESTRONICA en nuestro país, de SINCLAIR C5, que no es sino el triciclo con motor eléctrico, que el tío CLIVE ha inventado. Algunos detalles importantes, en Inglaterra, el precio oscila alrededor de las 80.000 pts., se sirve contra pedido por correo, y el tiempo que tardan en remitirlo es de unos 15 días, y ahora lo más importante, podrá este pequeñísimo y poco potente vehículo, con las pendientes españolas... tengamos en cuenta, que ha sido diseñado para llanear por las carreteras inglesas. Dejémoslo en interrogante.

QUEREMOS VER TUS PROGRAMAS BASIC

Sabemos que eres lo suficientemente ingenioso para no necesitar copiarlos de nadie. Mandanos una cinta de cassette con tu programa, y a ser posible, un listado del mismo.
Premiamos con 5.000 pts. y un ejemplar de la revista en cuestión, cada programa que publiquemos en cualquiera de nuestras cinco publicaciones.

¡¡SUSCRIBETE A 48K!!

De regalo recibirás el nº 1 del popular SOFTWARE MAGAZINE que incluye 2 fabulosas cassettes con programas estrella más la revista Super Juegos.

Solicito me inscriban como suscriptor de su revista por un año (12 entregas). 4.500 pts.

El importe lo abonaré de la siguiente forma:
Giro Postal nº........
- ☐ Contra reembolso
- ☐ Talón bancario a MONSER, S.A.
- ☐ c/ Argos, 9 - 28037 MADRID

Recorta o copia

Nombre ....
Dirección ....
Ciudad .... Provincia .... D.P. ....
Tlf. ....

En nuestra redacción todos somos humanos, por eso tenemos duendes, y como lo hemos descubierto, os lo presentamos.
Este interface programable, es el que debió aparecer en la foto del Análisis de Hardware de nuestro nº 1 pág. 10.
Perdonadle, sólo es un duende.

sinclair - spectrum
48-K

SOFTWARE MAGAZINE















Ya está a la venta el nº 7

sinclair - spectrum 48-K
PVP 725 ptas.
SOFTWARE MAGAZINE
CON ESTE NUMERO SIETE
2 SUPER CASSETTES «FULL MEMORY 48K»
CON INSTRUCCIONES EN CASTELLANO
MAS LA REVISTA SUPER JUEGOS
REVISTA DE APARICION MENSUAL

GOLF

Conviértete en un jugador de Golf e intenta acabar el recorrido de 18 hoyos en la menor cantidad de golpes posibles. Para iniciar el recorrido tienes un completo equipo de palos. Elige el adecuado para cada golpe, además del swing y la dirección apropiada. Pero ten cuidado con los espectadores, no les gusta que les golpee la bola y la pueden lanzar lejos del hoyo para molestarte.

FRUIT MACHINE

Esta es la popular máquina tragaperras. Hazte rico intentando hacer el Jack Pot. Esta máquina funciona igual que la de los bares y casinos, admitiendo la posibilidad de avances, detención de columnas, además de poder jugarte el premio obtenido, duplicándolo o perdiendo. Prueba tu suerte.

MONSER

Para envío
MONSER
c/ Argos,
28037 Madri
Teléf. 742 72 12/9

Y acaba de salir el nº 8 con: "ZIG-ZAG" y "JUMBLY"

Aparición mensual, cada número consta de 2 cassettes, más la Revista Super Juegos o Jakeka

¡NO TE LO PIERDAS!

DE VENTA EN KIOSCOS Y TIENDAS ESPECIALIZADAS



CODIGO BINARIO Y HEXADECIMAL

CAPITULO II

1. INTRODUCCION

Un lenguaje podríamos definirlo como el conjunto de signos, símbolos, palabras y métodos para la combinación de ellos, que tiene por finalidad posibilitar la comunicación entre dos entes.

Para comunicarnos con el ordenador no cabe duda que necesitamos un lenguaje. Pero el ordenador solo sabe interpretar las señales eléctricas que, a través de los buses, le llegan a los pines del microprocesador. Pero además, él no entiende de la diferencia que hay entre una tensión de 3,5 V y la de 4 V. El microprocesador simplifica estas señales, y todas las que vienen, en dos estados: hay, o no, tensión. No existen más matizaciones. También nosotros deberíamos acostumbrarnos a estos "estados" si queremos comunicarnos con él. Pero de algún modo tendremos que representar estas condiciones. Conceptualmente podemos hacerlas representables, mediante el encendido y apagado de una lámpara, la apertura y cierre del interruptor, etc. Hay que volver a insistir en que el ordenador no sabe más que sí o no.

Para representar estos dos estados se han elegido los números 0 y 1. Esto es lo que se denomina Sistema Binario.

2. BIT Y CODIGO BINARIO

Abreviatura de BInary digiT (Dígito binario) el BIT es la unidad elemental de información. Mediante un bit podemos conocer el estado de un ente. Si quisiéramos representar el estado de una lámpara, y tomáramos como norma el que un 1 va a indicar que está encendida, y un 0 indica que está apagada, cualquier interlocutor que, en lugar concertado para señalarlo, encontrase un 1 sabría sin lugar a dudas que la lámpara está encendida, y si encontrase un 0 sabría que está apagada. El lugar previamente acordado para señalar esa información es lo que denominaremos bit. Como se ve el bit sólo puede tomar los valores 0 y 1.

Con un bit solo podemos representar el estado de un elemento, su existencia o no. Ello representa muy poca información. Algo así como si en el sistema decimal sólo pudiéramos representar un número entero comprendido entre 0 y 9. También en el sistema binario, como en el decimal, podemos añadir dígitos a la izquierda, según queramos representar un número mayor, o vayan acrecentándose nuestras necesidades. Así si queremos representar hasta 4 posibles condiciones diferentes aumentaremos el número de dígitos binarios a 2. Veamos:

Orden	Condición	Código Binario
1	Ø	ØØ
2	1	01
3	2	1Ø
4	3	11

Del mismo modo podremos ir añadiendo al código binario para ir aumentando las posibilidades de representación.

Dado que nuestra mentalidad no está habituada a esta forma de trabajar, sería conveniente que nos acostumbrásemos. Lo primero que tendríamos que aprender es a transformar cualquier número decimal en su representación binaria. Para ello podríamos dar una regla muy sencilla:

"Dividir por dos el número decimal y sus cocientes sucesivos hasta que el cociente sea 1. El código binario estará compuesto por el último cociente y los restos que nos dieron las sucesivas divisiones en orden inverso a su obtención".

Trataremos de dejarlo más claro con un ejemplo. Supongamos el número 240 decimal y vamos a hallar su representación binaria. Según la regla anterior lo primero que tenemos que hacer es dividir por 2 el número y sus cocientes sucesivos:

```
Orden en
Código      240 | 2
 8 ........  Ø   120 | 2
 7 ............   Ø    60 | 2
 6 ................    Ø    30 | 2
 5 ....................     Ø    15 | 2
 4 ........................      1     7 | 2
 3 ..............................      1    3 | 2
 2 ....................................     1   1
 1 .............................................1
```

Luego se pone el último cociente seguido de los sucesivos restos que nos dieron las divisiones, en orden inverso a su obtención. Este número de orden lo hemos expuesto a la izquierda al hacer la división. El código será pues:

24Ø decimal = 1111ØØØØ binario

Para practicar esta regla de transformación vamos a ayudarnos del ordenador. Para ello bastará que introduzcas el siguiente programa, escrito en Basic, el cual te propondrá un número decimal y te pedirá su transformación a binario. Si la respuesta es correcta te lo señalará, pero en caso negativo te responde con el código correcto.

Este mismo programa puede servirnos para traducir a binario cualquier número decimal, con tal de que hagamos unas ligeras y sencillas modificaciones.

3. OCTETO, BYTE

Un solo bit no puede representar gran cantidad de información como ya hemos dicho, pero tampoco puede facilitar la comprensión, entre los fabricantes y usuarios de ordenadores, el que cada uno use de modo indiscriminado el número de bits que le apetezca, sin ninguna norma que lo regule. Para evitar el caos que de alguna forma podría avecinarse, se llegó a la conclusión de que lo mejor era agrupar los bits de ocho en ocho formando lo que denominamos **octeto o byte**.

El byte es la unidad de información que maneja el ordenador en todas las operaciones de movimiento, comparación, etc. como veremos más adelante. Esto no quita el que nosotros no podamos dirigirnos a un bit dentro del byte, pero siempre direccionaremos un byte.

En un byte puede representarse hasta 256 condiciones diferentes.

4. CODIGO HEXADECIMAL

Escribir, o leer, en la memoria del ordenador a base de utilizar el código binario, puede suponer una dificultad tan grande que el hacer un programa puede resultar poco menos que imposible. Para solucionar este problema se ideó la forma de utilizar menos dígitos, con lo que el comunicarse con el ordenador podría ser mucho más fácil. Esta solución consistía en la utilización del código hexadecimal, es decir, base 16.

Este sistema de numeración se compone de 16 signos correspondientes a los 10 del sistema decimal (de 0 a 9) a los que se les ha añadido las primeras letras del abecedario, A, B, C, D, E, F, que representan los número decimales 10, 11, 12, 13, 14, y 15 respectivamente.

De este modo, un byte, u octeto, que en código binario estaba representado por 8 dígitos, en el sistema hexadecimal lo está solamente por solo 2 dígitos.

Mediante el siguiente programa obtendrás una tabla de las equivalencias de los 256 primeros números decimales con sus correspondientes códigos en hexadecimal y binario:

```
   5 REM Progra
  10 REM Tabla
  20 CLS : PRIN
ec.  Binario"
  30 FOR x=0 TO
00000": LET a=X
  40 FOR y=8 TO
=INT (a/2): LET
b$(y)=CHR$ a: L
  50 LET a=x: L
LET a=a-16*b
  60 LET a=a+48
T a=a+7
  70 LET b=b+48
T b=b+7
  80 PRINT TAB
B 12;CHR$ b;CHR
  90 NEXT x
```

Existen unas tablas que pueden realizarse para hacer la traducción de decimal a hexadecimal de una forma más o menos sencilla. A continuación las reproduciremos para su uso en caso de necesidad.

No obstante, lo más cómodo es la utilización para esta tradución, en ambos sentidos, las opciones correspondientes del programa CARGAHEX.

MONTADO

```
  10 REM Programa DECIBIN
  20 CLS : PRINT TAB 8;"DECIMAL
a BINARIO","Decimal   Respuesta
  Solucion"
  30 LET a=INT (RND*256): LET c=
a: PRINT TAB 5-LEN STR$ a;a;: LE
T b$="00000000": FOR x=8 TO 1 ST
EP -1: LET b=INT (a/2): LET a=a-
2*b+48: LET b$(x)=CHR$ a: LET a=
b: NEXT x
  40 INPUT "Como es en binario ?
",a$
  50 IF LEN a$=0 OR LEN a$>8 THE
N GO TO 40
  60 FOR x=1 TO LEN a$: IF a$(x)
<>"1" AND a$(x)<>"0" THEN PRINT
"Solo se admite 0 y 1": GO TO 30
  70 NEXT x: PRINT TAB 18-LEN a$
;a$;: LET b=0: FOR x=1 TO LEN a$
: LET b=2*b+VAL a$(x): NEXT x
  80 IF b=c THEN PRINT TAB 22;"C
ORRECTO": GO TO 30
  90 PRINT TAB 22;b$: GO TO 30
```

Conversión Hexadecimal—Decimal

BYTE				BYTE				BYTE			
0123		4567		0123		4567		0123		4567	
HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC
0	0	0	0	0	0	0	0	0	0	0	0
1	1.048.576	1	65.536	1	4.096	1	256	1	16	1	1
2	2.097.152	2	131.072	2	8.192	2	512	2	32	2	2
3	3.145.728	3	196.608	3	12.288	3	768	3	48	3	3
4	4.194.304	4	262.144	4	16.384	4	1.024	4	64	4	4
5	5.242.880	5	327.680	5	20.480	5	1.280	5	80	5	5
6	6.291.456	6	393.216	6	24.576	6	1.536	6	96	6	6
7	7.340.032	7	458.752	7	28.672	7	1.792	7	112	7	7
8	8.388.608	8	524.288	8	32.768	8	2.048	8	128	8	8
9	9.437.184	9	589.824	9	36.864	9	2.304	9	144	9	9
A	10.485.760	A	655.360	A	40.960	A	2.560	A	160	A	10
B	11.534.336	B	720.896	B	45.056	B	2.816	B	176	B	11
C	12.582.912	C	786.432	C	49.152	C	3.072	C	192	C	12
D	13.631.488	D	851.968	D	53.248	D	3.328	D	208	D	13
E	14.680.064	E	917.504	E	57.344	E	3.584	E	224	E	14
F	15.728.640	F	983.040	F	61.440	F	3.840	F	240	F	15
6		5		4		3		2		1	

Un número binario de 8 bits se puede convertir fácilmente a hexadecimal. Para facilitarte esta tarea es conveniente dividir los números binarios en dos grupos de 4 bits cada uno. A continuación se sustituye cada uno de los grupos de 4 bits por el dígito equivalente en hexadecimal según la siguiente tabla:

Binario	Hexadecimal
ØØØØ	Ø
ØØØ1	1
ØØ1Ø	2
ØØ11	3
Ø1ØØ	4
Ø1Ø1	5
Ø11Ø	6
Ø111	7
1ØØØ	8
1ØØ1	9
1Ø1Ø	A
1Ø11	B
11ØØ	C
11Ø1	D
111Ø	E
1111	F

Por ejemplo, consideremos binario de 8 bits:

11ØØ11Ø1

Primeramente, lo separamos en dos grupos de cuatro dígitos binarios cada uno.

11ØØ   11Ø1

Finalmente, sustituiremos uno de los grupos por su equivalente hexadecimal en la tabla anterior.

CD

Es decir, el número binario 11001101 se puede representar como CD en hexadecimal.

En adelante, y para distinguir un número decimal del hexadecimal, y dado que los compiladores más utilizados emplean la misma notación, procuraremos anteponer el signo # para así evitar confusiones.

CAPITULO III

ARQUITECTURA DEL MICROPROCESADOR

1. INTRODUCCION

Se entiende por arquitectura de un microprocesador aquellos elementos que constituyen internamente el micro y que definen las características que lo hacen más o menos fácil de programar.

Entre estos elementos se encuentran los registros de uso general, los acumuladores, el stack o pila, etc.

Existen diversos tipos de microprocesadores que, en lo que a nosotros respecta, podríamos agruparlas según la longitud de palabra: 1, 2, 4, 8, 16 ó 32 bits.

la palabra es la unidad de tratamiento de información. Su longitud es el número de bits en paralelo que son tratados simultáneamente en cada instrucción del microprocesador.

Pues bien, el Z-80 es un microprocesador de 8 bits y además de ser de los más conocidos, es de los más rápidos y potentes. La velocidad media de ejecución de una instrucción es de 3 μs y el número de instrucciones diferentes que posee es de 138. Está fabricado en tecnología NMOS, lo cual representa una economía en el consumo de corriente.

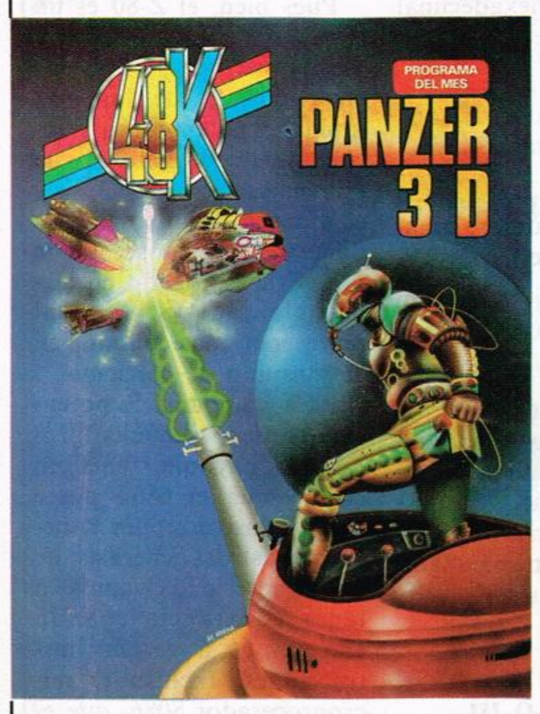
Para comparar la potencia que tiene este microprocesador hay que tener en cuenta que el 6502 posee solo 57 instrucciones y menos registros generales de cuya necesidad vamos a tener constancia cuando empecemos a ver las instrucciones. Incluso el microprocesador 8086, que es el que equipa a los ordenadores IBM/PC y compatibles, tiene una longitud de palabra de 16 bits, posee sólo 78 instrucciones y el tiempo medio de ejecución es de 800 μs.

2. ARQUITECTURA DEL PROCESADOR CENTRAL

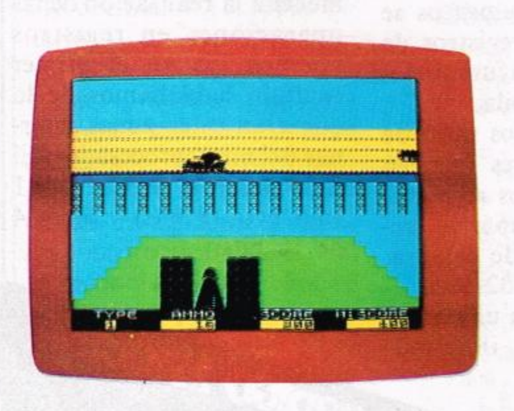
El Z-8Ø es un microordenador orientado principalmente a la realización de las operaciones en registros internos (ya en el primer capítulo hablábamos de lo que eran estas zonas internas del microprocesador) El Z-8Ø posee 18 registros de 1 solo byte (8 bits) y otros 4 de 16 bits.



PANZ



Nuestro espía tras las líneas enemigas, nos hizo saber por el conducto habitual, la proximidad del desembarco enemigo; el día D estaba cercano, la noche cerrada con el cielo totalmente nublado, no permitía ver ni un alma ni una estrella en todo el universo, el mensajero me avisaba de que el comandante en jefe tenía algo que decirme, abandoné mi tienda de campaña, y acompañé al sargento mayor, que me condujo a presencia del comandante.



Teniente, me dijo, hemos tenido noticias de Herbie (Herbie era el nombre clave de nuestro espía), mañana, al amanecer, y a pocos kilómetros de aquí, desembarcará el grueso de la fuerza enemiga, los tanques PANZER, llegarán a tierra, a través de un puente que su servicio de ingenieros ha diseñado, y que será instalado unas horas antes, su misión consistirá en emplazar nuestra artillería, y conseguir que ni un solo tanque enemigo llegue a tierra...

Muy bien podía ser esta la historia que diera comienzo a nuestro juego, pues de eso se trata, de evitar que los tanques que atraviesan el puente, lleguen a su destino.

LOS TANQUES

Avanzan siempre en línea recta; pueden disparar su cañones, o bien hacia nosotros, lo que sabremos cuando giren la torreta, (será un aviso de que nos apartemos), o bien hacia adelante, para limpiar su camino de los restos de chatarra de los tanques que nosotros hayamos abatido con nuestra artillería pesada.

EL PUENTE

Simula un puente ligero de ingenieros, tiene cuatro carriles por los que aparecerán los tanques enemigos, en cuatro niveles distintos.

NUESTRA ARTILLERIA

Consiste en un cañón, con guía de altura de tiro de cinco posiciones, una para cada nivel de carriles del puente, y otra para... más adelante veremos.



El movimiento a derecha e izquierda será con la teclas "T" y "F" respectivamente, la inclinación del cañón la controlaremos con las teclas "W" para subir y "S" para bajar, y el disparo, con la "Y", igualmente, disponemos de una tecla para tomarnos un respiro, la "H", que detendrá momentáneamente el juego, para que recobremos el resuello.

ESTRATEGIA

Tendremos que manejar nuestra arma, de manera que acertemos de lleno al enemigo, de lo contrario simplemente le inmovilizaremos, y será necesario un segundo disparo mucho más certero, para abatirlo totalmente. El punto clave, es la torreta, y la alcanzaremos, disparando un punto por encima del que le correspondería, dado el carril que ocupe, pero teniendo cuidado, pues cuando menos lo esperemos nos devolverá el fuego, acabando así la partida.

ER 3D

GENERALIDADES

Al comienzo de cada partida, se nos ofrece un menú de cuatro posibilidades, fácilmente comprensibles, en la ''I'' nos ofrece información del juego. Con la ''C'' cambiaremos las características del mismo: rapidez de disparo, racionamiento de la munición, etc..., y finalmente la redefinición de teclas, para una más ergonómica distribución de las mismas.

En general es un buen juego, muy entretenido, y con un gran abanico de posibilidades con las que os aseguro unas horas de inmejorable divertimento.



Dentro del primer grupo existen dos subconjuntos de 8 registros cada uno que se denominan principal y alternativo.

Cada uno de estos subconjuntos está compuesto por 8 registros de 8 bits, de uso general, y que pueden ser utilizados de forma individual o bien formando pares con otro.

A cada uno de los registros se les ha designado una letra, o dos, que los identifica en la descripción de las instrucciones.

Así, al grupo de registro principal se les nombra con las letras: A, F, B, C, D, E, H, y L y a los correspondientes del grupo alternativo se les denomina con idénticas letras pero añadiendo un apóstrofe (A', F', B', C', E', H' y L'). En la figura 1 encontrarás todos los registros existentes y su denominación.

Por supuesto, la arquitectura del Z-80 no está compuesta únicamente por los registros, aunque sea lo único en que vamos a detenernos. También existen:

— Unidad de Decodificación de instrucciones.

— Unidad de control del bus de datos

— Unidad de control del bus de direcciones

— Unidad Aritmético-lógica (ALU)

3. REGISTROS

Vamos ahora a detenernos en la descripción de los registros internos de la CPU:

—Acumuladores (A y A')

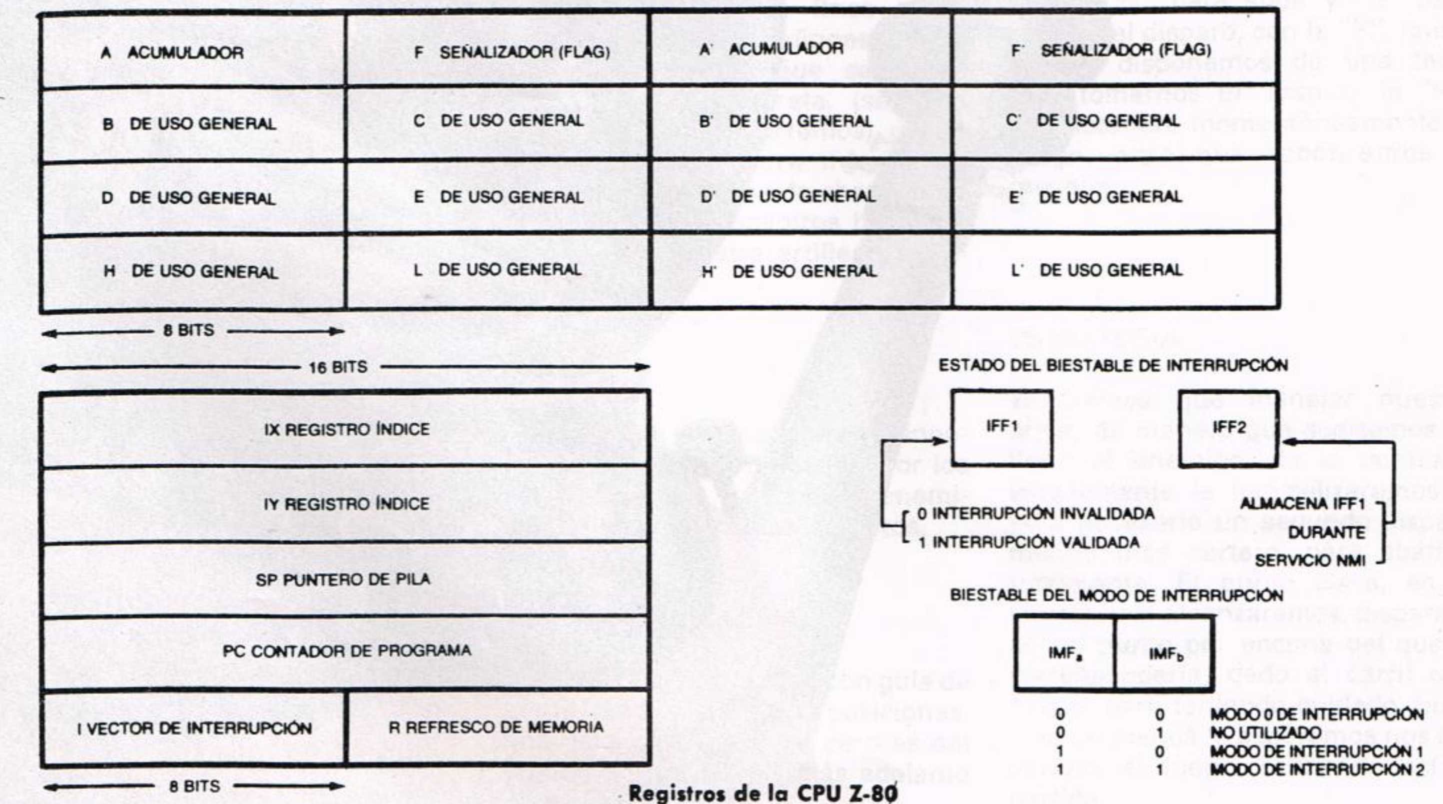
El procesador central (CPU) tiene dos registros Acumuladores de 8 bits: El Principal y el Alternativo. En el Acumulador principal es donde se almacenan los resultados de los cálculos intermedios y finales de las operaciones aritméticas. Hay que tener en cuenta que, como más adelante veremos las operaciones solo se realizan sobre el principal, pero el programador puede seleccionar cual es en cada momento el principal.

— Registros de estado (F y F')

La CPU tiene dos registros de estado también, como en el caso de los Acumuladores. En el principal deja señalada la ocurrencia de condiciones específicas, lógicas o aritméticas, producidas en el procesador como consecuencia de la ejecución de una instrucción, como puede ser el desbordamiento en una suma, resultado negativo en una resta, o que el acumulador tiene un resultado cero, etc.

— Registros de uso general (B, C, D, E, H, L y B', C', D', E', H' y L')

Como ya dijimos en el párrafo anterior, el Z80 está orientado a que las instrucciones se realicen sobre registros. Pues bien existen unos registros sobre los que recaen casi todo el peso de las operaciones que puede realizar el microprocesador. Uno de esos registros es el Acumulador principal, y los otros son los que denominamos "registros de uso general" porque su utilización no se basa en una especialización similar a la que tiene adjudicada otros registros, por ejemplo, la ya señalada para el Acumulador.



Registros de la CPU Z-80

Hay dos conjuntos de registros de uso general: Principal y Alternativo. Cada uno de dichos conjuntos está formado por seis registros de un solo byte. En todas las instrucciones se utilizan los registros del grupo principal. Unicamente existe una instrucción que permite la permuta de datos entre los registros de ambos conjuntos.

Los registros de este grupo intervienen en instrucciones de 16 bits, comportándose como registros de 16 bits. Para ello se agrupan en pares (BC, BE, HL) formando tres registros de 16 bits. No pueden formarse pares diferentes a los aquí especificados.

— Registros de uso especial.

Dentro de este grupo vamos a ver aquellos que tienen una misión específica dentro del funcionamiento del microprocesador.

— Registro contador del programa (PC)

Este registro, de 16 bits, contiene la dirección de memoria en que se encuentra la próxima instrucción a ejecutarse. Sucede que tras la ejecución de una instrucción, el contenido de este registro se incrementa en una cantidad igual a la longitud de la instrucción ejecutada. Esta regla se exceptúa en el caso de que la instrucción ejecutada sea de salto a una dirección o en las llamadas subrutinas en cuyo caso tomará el valor de la dirección de memoria en que se encuentra la instrucción en la que debe continuar la ejecución del programa.

— Puntero de pila o stack (SP).

Para comprender la función de este registro debemos explicar lo que es una pila o stack. El stack es una zona de memoria reservada para el uso del microprocesador en donde unas instrucciones van dejando, y otras van cogiendo, información en grupos de dos bytes. Esta información es siempre el contenido de los registros pares, o el contenido del registro PC, cuando se hace una llamada a subrutinas.

El registro SP, de 16 bits, guarda la dirección de la última entrada, o dato, que ha sido almacenado en el stack. La norma que rige el almacenamiento de información en la pila y su posterior recuperación es la siguiente:

"Ultimo en entrar, primero en salir"

Más claramente expresada, esta norma indica que cuando se introduce un dato en el stack, el registro SP pasa a apuntar la dirección de este dato. A la hora de pedir una información al stack, este nos dará la que esté apuntada por este registro que a su vez pasará a direccionar el dato que fue introducido con anterioridad al sacado.

Una peculiaridad del funcionamiento de stack es su forma de llevar las anotaciones. Cuando se introduce un dato este se guarda en la dirección marcada por el SP el cual se decrementa en dos posiciones previamente. A la hora de recuperar la información, la instrucción obtiene la misma del stack y deja al registro SP apuntado a dos bytes más altos que la anterior dirección.

Por este motivo, el stack se sitúa en las posiciones más altas de memoria, como si dejásemos toda la memoria como zona de pila.

— Registro de interrupciones (I)

Una señal externa que le llegue a Z8Ø a través del pin INT puede hacer que el programa, que en ese momento se está ejecutando, se interrumpa. Esto dependerá de si lo hemos permitido o no mediante unas instrucciones de control. Pero cuando se interrumpe la ejecución del programa, el control debe ser cedido a alguna dirección de memoria. Cuando la interrupción hace que el control se ceda a una posición de memoria por encima de los 256 bytes, dicha dirección debe estar contenida en dos bytes. Uno de esos bytes es el contenido de este registro I. El dispositivo que provoca la interrupción debe proveer el otro byte que forma la dirección de salto. Esto permite que las rutinas de interrupción sean colocadas dinámicamente en cualquier lugar de la memoria, facilitando el acceso a las mismas.

— Registro índice (IX y IY)

Estos registros de 16 bits son utilizados en las instrucciones de direccionamiento indexado. Se utilizan como base para apuntar a zona de memoria que tenga definida una estructura similar a la de una tabla. En conjunción con el byte adicional que se incluye en las instrucciones que hacen uso de estos registros permiten el manejo de los datos de un modo fácil y cómodo.

— Registro de refresco (R)

También llamado de regeneración de memorias dinámicas permite la utilización de este tipo de hardware con la misma facilidad que si de memorias estáticas se tratase.

Las memorias dinámicas son los dispositivos hardware que almacenan, en casi todos los microprocesadores, la información que se encuentra en RAM. Son unos dispositivos que por su tecnología permiten almacenar más información en menor espacio, y son más baratas, pero por contra tienen el defecto de que la información contenida en ellas, al cabo de un tiempo muy corto, desaparece si no es "refrescada". Para esta función de refresco es para lo que se utiliza este registro.

Siete bits de este registro son incrementados después de la fase de búsqueda de la instrucción. El octavo bit permanece con el mismo valor que tuviera, y solo será modificado por una instrucción de carga del acumulador en ese registro.

Todo el método de refrescar la memoria es totalmente transparente al programador y no retrasa la ejecución de las instrucciones.

Aunque este registro pudiera utilizarse como registro general no es aconsejable su uso por parte del programador.

En el próximo capítulo abordaremos los diversos tipos de instrucciones y veremos en profundidad, y de modo práctico, algunas de ellas, como son las de retorno y la carga de un valor en los registros.

C-30 CASSETTE ESPECIAL PARA ORDENADOR



Con la marca Monser sobre el cassette, usted obtiene no solamente una excelente cinta para computador, sino también una cassette que le proporciona todas las funciones y conveniencias que requiera el usuario. El cassette para ordenadores personales Monser está diseñado para ser usado con microcomputadores y provee una combinación única de precisión y ejecución.

DE VENTA EN TIENDAS ESPECIALIZADAS.

Para envios dirigirse a Monser, S.A. c/ Argos nº 9. Tlf. 742 72 12 / 96.

CARGAHEX

MONTADO

INTRODUCCION

ESTE PROGRAMA FUE DISEÑADO Y REALIZADO CON LA PRINCIPAL MISION DE SERVIR DE AYUDA PARA LA MANIPULACION DE BYTES EN MEMORIA, YA FUERON FORMANDO PARTE DE UN PROGRAMA DE LENGUAJE MAQUINA PREVIAMENTE ENSAMBLADO A MANO, O COMO BLOQUES DE DATOS.
SEGUN FUERON CRECIENDO LAS NECESIDADES QUE TENIA DURANTE LA UTILIZACION DEL MISMO, ME LLEVO A AÑADIR NUEVAS OPCIONES QUE SATISFACIERAN DICHAS NECESIDADES.

INSTRUCCIONES

Una vez tecleado el programa, y oportunamente salvado en cinta, al teclear RUN el programa tiene que empezar a funcionar sin ningún problema. De no ser así convendrá revisar el listado del mismo comprobándolo con el que ofrecemos.

Al comienzo del programa aparece el Menú con las opciones, las cuales vamos a comentar.

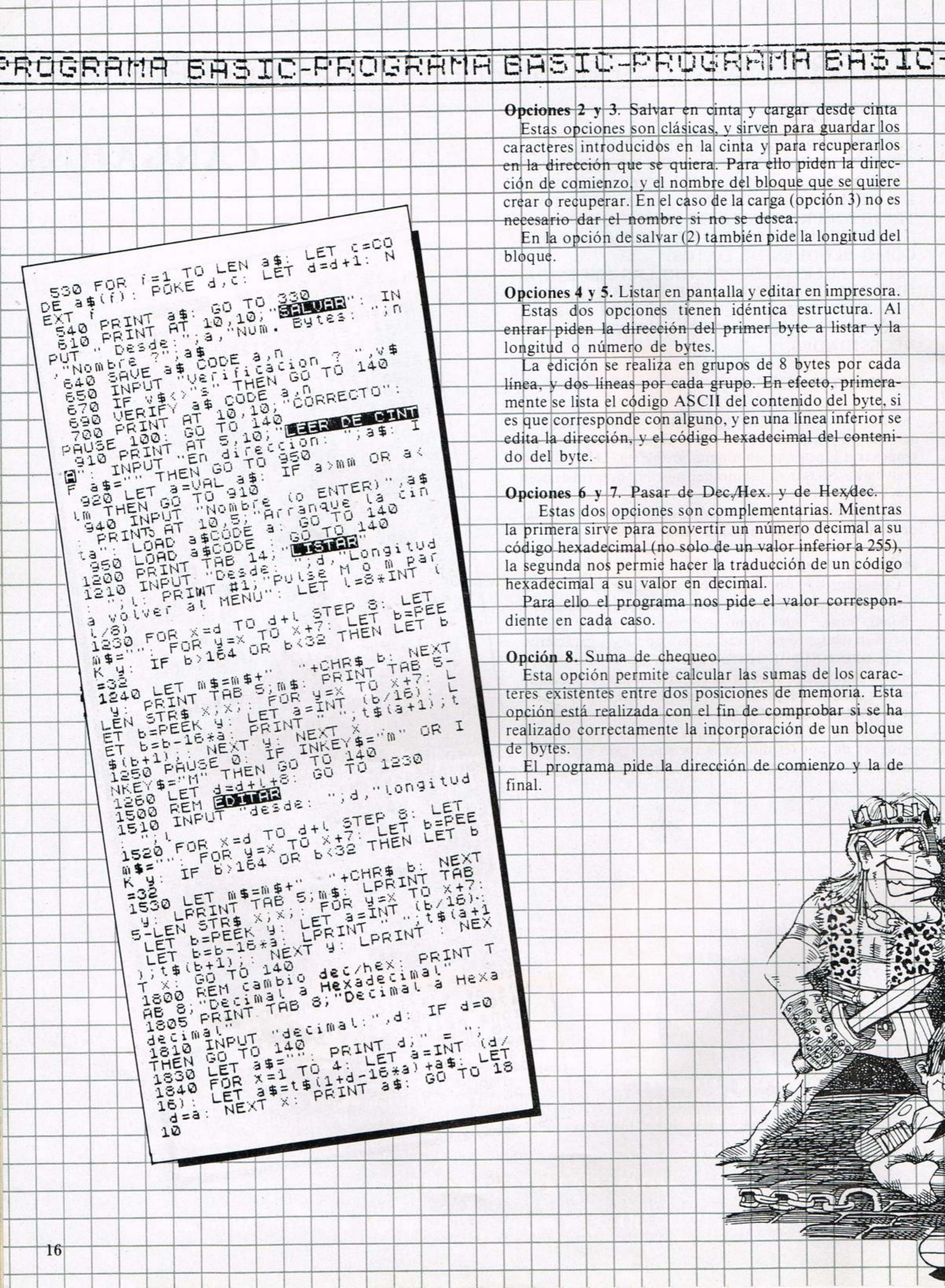
Opción 1. Escribir en memoria

Con esta opción se puede escribir, a partir de la dirección que se quiera, dada en decimal y superior a 30.000, tantos bytes como se deseen, los cuales pueden introducirse tanto en hexadecimal, como en caracteres.

En principio la introducción se realizará en hexadecimal, pero si se pulsa ENTER sin haber tecleado ningún caracter, el programa pide que se le den los datos en formato "caracter".

Si no se le da ningún dato en formato caracter entonces da por terminada la incorporación y presenta en pantalla el número de bytes introducidos.

```
  10 REM Programa CARGAHEX por B
lanca Prieto
  20 LET a=0: LET RT=PEEK 23732+
256*PEEK 23733
  21 LET t$="0123456789ABCDEF"
  22 LET mm=65347: LET UDG=65367
  24 LET ni=INT (UDG/256): POKE
23675,UDG-ni*256: POKE 23676,ni
 100 LET lm=30000
 140 REM Menu principal
 150 CLS : PRINT AT 2,10;"MENU";
AT 4,3;"1.- Escribir en memoria
";AT 5,3;"2.- Salvar en cinta";
AT 6,3;"3.- Cargar desde cinta"
;AT 7,3;"4.- Listar en pantalla
";AT 8,3;"5.- Editar por impres
.";AT 9,3;"6.- Pasar de Dec./He
x.";AT 10,3;"7.- Pasar de Hex./
Dec.";AT 11,3;"8.- Suma de cheq
ueo";AT 12,3;"9.- Pasar de dec.
a Exp.";AT 13,3;"10.- Pasar de
EXP. a Dec.";AT 14,3;"11.- Mover
zonas de mem.";AT 15,3;"12.- Pr
obar rutinas";AT 20,1; FLASH 1;"
OPCION"
 210 INPUT LINE a$: IF a$="" THE
N GO TO 210
 220 LET a=VAL a$: IF a<0 OR a>1
2 THEN GO TO 210
 240 CLS : GO TO 300*a
 300 PRINT TAB 10;"ESCRIBIR","Pr
imera direccion libre = ";lm,"Es
cribe en ?";
 310 INPUT d: IF d>mm OR d<lm TH
EN GO TO 310
 320 PRINT d: LET h=d
 330 LET a$=""
 340 IF a$="" THEN PRINT "codigo
en hexadecimal": INPUT a$
 350 IF a$="" THEN GO TO 520
 360 IF a$(1)="m" OR a$(1)="M" T
HEN GO TO 490
 370 IF LEN a$/2<>INT (LEN a$/2)
THEN PRINT "error": GO TO 330
 390 FOR f=1 TO LEN a$ STEP 2
 400 LET a=CODE a$(f)-48: IF a>9
THEN LET a=a-7
 401 IF a>16 THEN LET a=a-32
 410 LET b=CODE a$(f+1)-48: IF b
>9 THEN LET b=b-7
 411 IF b>16 THEN LET b=b-32
 420 PRINT a$(f TO f+1);" ";
 430 LET c=16*a+b: POKE d,c: LET
d=d+1
 460 IF d=UDG THEN PRINT "Ojo: z
ona de graficos"
 470 IF d=UDG-20 THEN PRINT "Ojo
:zona peligrosa"
 480 NEXT f: PRINT : GO TO 330
 490 PRINT : PRINT "Memoria escr
ita: ";d-h: PAUSE 100: GO TO 140
 520 PRINT "Datos en caracteres
": INPUT a$: IF a$="" THEN GO TO
490
```



```
530 FOR f=1 TO LEN a$: LET c=CO
DE a$(f): POKE d,c: LET d=d+1: N
EXT f
540 PRINT a$: GO TO 330
610 PRINT AT 10,10;"SALVAR": IN
PUT "Desde:";a,"Num. Bytes: ";n
,"Nombre ?";a$
640 SAVE a$ CODE a,n
650 INPUT "Verificacion ? ",v$
670 IF v$<>"s" THEN GO TO 140
690 VERIFY a$ CODE a,n
700 PRINT AT 10,10;"CORRECTO":
PAUSE 100: GO TO 140
910 PRINT AT 5,10;"LEER DE CINT
A": INPUT "En direccion: ";a$: I
F a$="" THEN GO TO 950
920 LET a=VAL a$: IF a>mm OR a<
lm THEN GO TO 910
940 INPUT "Nombre (o ENTER)",a$
: PRINT AT 10,5;"Arranque la cin
ta": LOAD a$CODE a: GO TO 140
950 LOAD a$CODE : GO TO 140
1200 PRINT TAB 14;"LISTAR"
1210 INPUT "Desde: ";d,"Longitud
: ";l: PRINT #1;"Pulse M o m par
a volver al MENU": LET l=8*INT (
l/8)
1230 FOR x=d TO d+l STEP 8: LET
m$="": FOR y=x TO x+7: LET b=PEE
K y: IF b>164 OR b<32 THEN LET b
=32
1240 LET m$=m$+" "+CHR$ b: NEXT
y: PRINT TAB 5;m$: PRINT TAB 5-
LEN STR$ x;x;: FOR y=x TO x+7: L
ET b=PEEK y: LET a=INT (b/16): L
ET b=b-16*a: PRINT " ";t$(a+1);t
$(b+1);: NEXT y: NEXT x
1250 PAUSE 0: IF INKEY$="m" OR I
NKEY$="M" THEN GO TO 140
1260 LET d=d+l+8: GO TO 1230
1500 REM EDITAR
1510 INPUT "desde: ";d,"longitud
: ";l
1520 FOR x=d TO d+l STEP 8: LET
m$="": FOR y=x TO x+7: LET b=PEE
K y: IF b>164 OR b<32 THEN LET b
=32
1530 LET m$=m$+" "+CHR$ b: NEXT
y: LPRINT TAB 5;m$: LPRINT TAB
5-LEN STR$ x;x;: FOR y=x TO x+7:
LET b=PEEK y: LET a=INT (b/16):
LET b=b-16*a: LPRINT " ";t$(a+1
);t$(b+1);: NEXT y: LPRINT : NEX
T x: GO TO 140
1800 REM cambio dec/hex: PRINT T
AB 8;"Decimal a Hexadecimal"
1805 PRINT TAB 8;"Decimal a Hexa
decimal"
1810 INPUT "decimal:",d: IF d=0
THEN GO TO 140
1830 LET a$="": PRINT d;" = ";
1840 FOR x=1 TO 4: LET a=INT (d/
16): LET a$=t$(1+d-16*a)+a$: LET
d=a: NEXT x: PRINT a$: GO TO 18
10
```

Opciones 2 y 3. Salvar en cinta y cargar desde cinta

Estas opciones son clásicas, y sirven para guardar los caracteres introducidos en la cinta y para recuperarlos en la dirección que se quiera. Para ello piden la dirección de comienzo, y el nombre del bloque que se quiere crear o recuperar. En el caso de la carga (opción 3) no es necesario dar el nombre si no se desea.

En la opción de salvar (2) también pide la longitud del bloque.

Opciones 4 y 5. Listar en pantalla y editar en impresora.

Estas dos opciones tienen idéntica estructura. Al entrar piden la dirección del primer byte a listar y la longitud o número de bytes.

La edición se realiza en grupos de 8 bytes por cada línea, y dos líneas por cada grupo. En efecto, primeramente se lista el código ASCII del contenido del byte, si es que corresponde con alguno, y en una línea inferior se edita la dirección, y el código hexadecimal del contenido del byte.

Opciones 6 y 7. Pasar de Dec./Hex. y de Hex/dec.

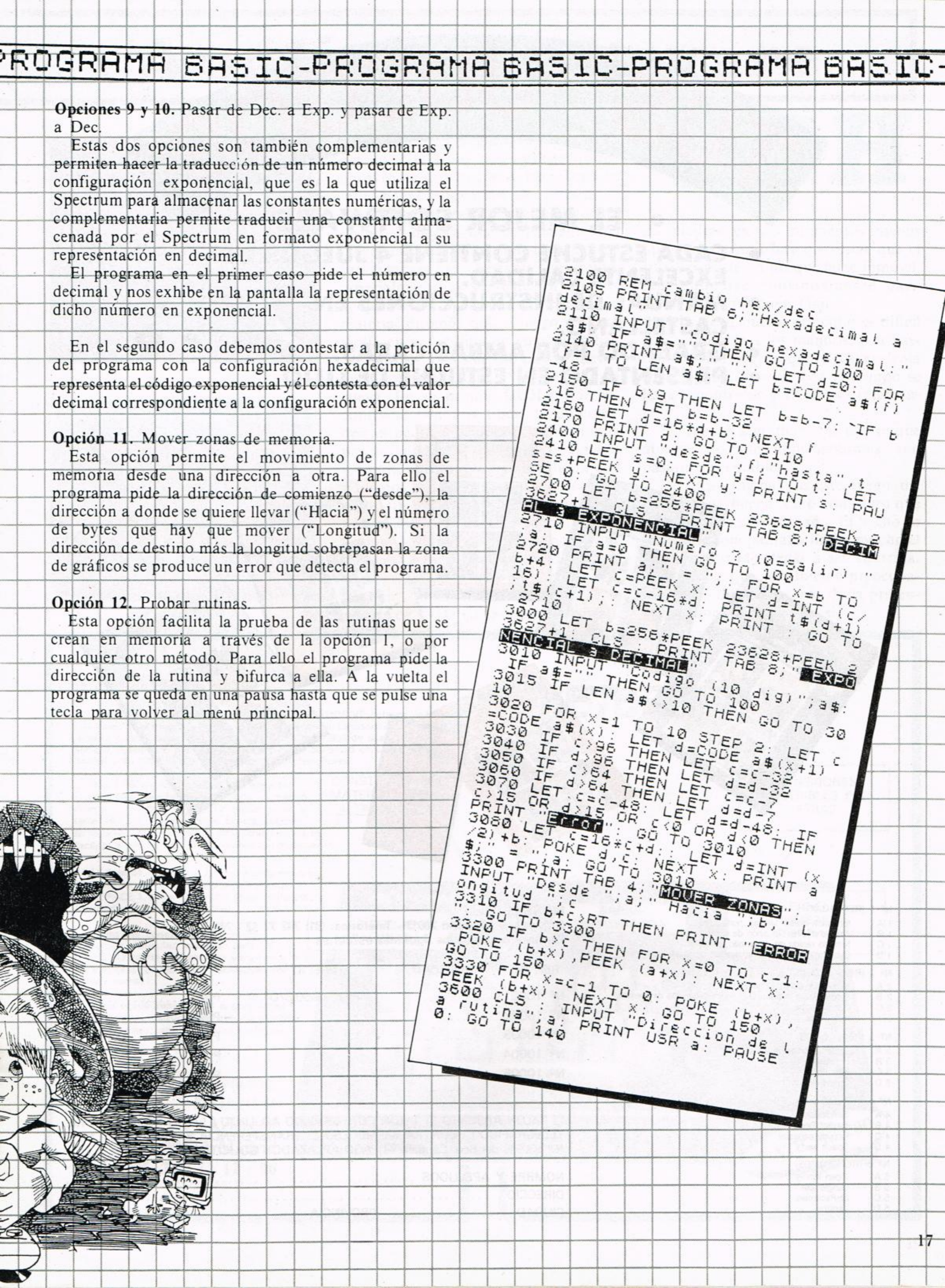
Estas dos opciones son complementarias. Mientras la primera sirve para convertir un número decimal a su código hexadecimal (no solo de un valor inferior a 255), la segunda nos permie hacer la traducción de un código hexadecimal a su valor en decimal.

Para ello el programa nos pide el valor correspondiente en cada caso.

Opción 8. Suma de chequeo.

Esta opción permite calcular las sumas de los caracteres existentes entre dos posiciones de memoria. Esta opción está realizada con el fin de comprobar si se ha realizado correctamente la incorporación de un bloque de bytes.

El programa pide la dirección de comienzo y la de final.



Opciones 9 y 10. Pasar de Dec. a Exp. y pasar de Exp. a Dec.

Estas dos opciones son también complementarias y permiten hacer la traducción de un número decimal a la configuración exponencial, que es la que utiliza el Spectrum para almacenar las constantes numéricas, y la complementaria permite traducir una constante almacenada por el Spectrum en formato exponencial a su representación en decimal.

El programa en el primer caso pide el número en decimal y nos exhibe en la pantalla la representación de dicho número en exponencial.

En el segundo caso debemos contestar a la petición del programa con la configuración hexadecimal que representa el código exponencial y él contesta con el valor decimal correspondiente a la configuración exponencial.

Opción 11. Mover zonas de memoria.

Esta opción permite el movimiento de zonas de memoria desde una dirección a otra. Para ello el programa pide la dirección de comienzo ("desde"), la dirección a donde se quiere llevar ("Hacia") y el número de bytes que hay que mover ("Longitud"). Si la dirección de destino más la longitud sobrepasan la zona de gráficos se produce un error que detecta el programa.

Opción 12. Probar rutinas.

Esta opción facilita la prueba de las rutinas que se crean en memoria a través de la opción 1, o por cualquier otro método. Para ello el programa pide la dirección de la rutina y bifurca a ella. A la vuelta el programa se queda en una pausa hasta que se pulse una tecla para volver al menú principal.

```
2100 REM cambio hex/dec
2105 PRINT TAB 6;"Hexadecimal a
decimal"
2110 INPUT "codigo hexadecimal:"
,a$: IF a$="" THEN GO TO 100
2140 PRINT a$;" ";: LET d=0: FOR
f=1 TO LEN a$: LET b=CODE a$(f)
-48
2150 IF b>9 THEN LET b=b-7: IF b
>16 THEN LET b=b-32
2160 LET d=16*d+b: NEXT f
2170 PRINT d: GO TO 2110
2400 INPUT "desde",f,"hasta",t
2410 LET s=0: FOR y=f TO t: LET
s=s+PEEK y: NEXT y: PRINT s: PAU
SE 0: GO TO 2400
2700 LET b=256*PEEK 23628+PEEK 2
3627+1: CLS : PRINT TAB 8;"DECIM
AL a EXPONENCIAL"
2710 INPUT "Numero ? (0=salir) "
;a: IF a=0 THEN GO TO 100
2720 PRINT a;" = ";: FOR x=b TO
b+4: LET c=PEEK x: LET d=INT (c/
16): LET c=c-16*d: PRINT t$(d+1)
;t$(c+1);: NEXT x: PRINT : GO TO
 2710
3000 LET b=256*PEEK 23628+PEEK 2
3627+1: CLS : PRINT TAB 8;"EXPO
NENCIAL a DECIMAL"
3010 INPUT "Codigo (10 dig)";a$:
 IF a$="" THEN GO TO 100
3015 IF LEN a$<>10 THEN GO TO 30
10
3020 FOR x=1 TO 10 STEP 2: LET c
=CODE a$(x): LET d=CODE a$(x+1)
3030 IF c>96 THEN LET c=c-32
3040 IF d>96 THEN LET d=d-32
3050 IF c>64 THEN LET c=c-7
3060 IF d>64 THEN LET d=d-7
3070 LET c=c-48: LET d=d-48: IF
c>15 OR d>15 OR c<0 OR d<0 THEN
PRINT "Error": GO TO 3010
3080 LET c=16*c+d: LET d=INT (x
/2)+b: POKE d,c: NEXT x: PRINT a
$;" = ";a: GO TO 3010
3300 PRINT TAB 4;"MOVER ZONAS":
INPUT "Desde ";a;" Hacia ";b;" L
ongitud ";c
3310 IF b+c>RT THEN PRINT "ERROR
": GO TO 3300
3320 IF b>c THEN FOR x=0 TO c-1:
 POKE (b+x),PEEK (a+x): NEXT x:
GO TO 150
3330 FOR x=c-1 TO 0: POKE (b+x),
PEEK (b+x): NEXT x: GO TO 150
3600 CLS : INPUT "Direccion de l
a rutina";a: PRINT USR a: PAUSE
0: GO TO 140
```



- **EL MEJOR SOFTWARE**
- **CADA ESTUCHE CONTIENE 4 JUEGOS DE EXCELENTE CALIDAD.**
- **MANUAL DE INSTRUCCIONES EN CASTELLANO.**
- **GRABADO POR AMBAS CARAS.**
- **PRESENTADO EN ESTUCHE DE LUJO.**

Nº 1 (REF. 10.001)
1 A Incursión aérea. Misiles, portaviones.
1 B Ataque misiles. Tu avión de combate.
1 C Método rápido de desarrollo quinielas.
1 D Gusano loco. Monstruos, arañas, disparos, láser, etc.

Nº 2 (REF. 10.002)
2 A Simulador de vuelo nocturno.
2 B Ajedrez para maestros. El mejor programa de ajedrez.
2 C Cosmonauta.
2D Arácnidos.

Nº 3 (REF. 10.003)
3 A Tesoro submarino.
3 B Fumigator.
3 C Mototron.
3 D Space Rocus.

Nº 4 (REF. 10.004)
4 A Simulador de vuelo F-18.
4 B Basket Bear.
4 C Convoy espacial.
4 D Space war.

Nº 5 (REF. 10.005)
5 A Open tennis.
5 B Jungle axe.
5 C S. Packmen.
5 D Jarama.

MONSER S.A.
C/Argos, 9 - Madrid 28017 - Teléfonos: (91) 742 72 12 - 742 72 96.
Por favor envíenme los siguientes estuches

REF. N.º	CANTIDAD	PRECIO	
Nº 10001	....	Ptas. 2.950 C/U	Ptas. ....
Nº 10002	....		Ptas. ....
Nº 10003	....		Ptas. ....
Nº 10004	....		Ptas. ....
Nº 10005	....		Ptas. ....
			Total ....

☐ TALON ADJUNTO ☐ TALON CONFORMADO ADJUNTO ☐ GIRO POSTAL ☐GIRO TELEGRAFICO ☐ CONTRA REEMBOLSO ☐ TRANSFERENCIA BANCARIA ☐ (Cta. N.º 836940 del Bco. Central). ☐ PAGO APLAZADO - SOLICITE INFORMACION

NOMBRE Y APELLIDOS ....
DIRECCION ....
CIUDAD .... PROVINCIA .... TEL. ....

UGH





Estamos ante un juego de cavernícolas. Se acerca la glaciación y nuestro amigo UGH se está preparando, aprovisionando su cueva. Este juego da muy buena impresión de entrada, y pienso que ha trabajado más el que ha hecho la música que el diseñador del juego. Esto lo digo porque la música de presentación es bastante sugerente, especialmente si se tiene un altavoz acoplado al Spectrum. Entrando en materia, diremos que el juego se desarrolla en una colina en cuya cima hay un nido de Pterodáctilo cuyos huevos debe recoger UGH y llevarlos a la cueva. En su misión, UGH puede encontrarse con dos tipos de dificultades: el pequeño dinosaurio que pulula por la colina, que tiene que ser evitado y contra el cual tenemos una lanza que podemos usar en caso de necesidad, pero, sin embargo, una vez que llegamos al nido solamente podemos llevar encima o el huevo o la lanza, por lo que la vuelta a la cueva se tiene que hacer desarmado. Pero aquí no acaba todo, ya que el Pterodáctilo intenta que la misión de UGH no se complete lanzando piedras en los momentos menos oportunos. Aquí hay que decir que el truco consiste en no quedarse inmóvil, sino moverse continuamente para evitar ser un blanco fácil.

Cabe decir que al principio es difícil acostumbrarse a los mandos, y es bastante difícil lograr que el cavernícola suba o baje, aunque con el tiempo se llega a controlar a UGH perfectamente.

En cuanto a gráficos, me parece que aunque no son impresionantes, son bastante aceptables.

La idea original de este juego me parece muy buena, pero considero que ha sido poco aprovechada. El hecho de tener una sola pantalla limita mucho el juego, pudiendo llegar a ser aburrida. Casi parece que se hubiesen preocupado, como dije antes, más de la presentación que del juego en sí.